

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: Информационный сайт об астероидах и
метеоритах на основе NASA API

Студент гр. 8303	_____	Почаев Н.А.
Студент гр. 8303	_____	Перелыгин Д.С.
Студент гр. 8304	_____	Алтухов А.Д.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург
2021

ЗАДАНИЕ
НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студенты

Почаев Н.А.

Перелыгин Д.С.

Алтухов А.Д.

Группа 8303 и 8304

Тема работы: Информационный сайт об астероидах и метеоритах на основе NASA API

Исходные данные:

Создание веб-сайта, предоставляющего функциональность по просмотру актуальных данных по метеоритам и астероидам, предоставляемых NASA API. Данные представляются в виде наглядных карт с отметкой мест падения, таблиц с полной известной информацией о небесном теле. Также по предоставленным данным возможен просмотр статистики в виде графиков. Имеется функционал добавления и импорта / экспорта данных из БД.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарии использования»

«Модель данных»

«Разработка веб-сайта»

«Выводы»

«Приложение»

Предполагаемый объем пояснительной записи:

Не менее 15 страниц.

Дата выдачи задания: 01.09.2020

Дата сдачи реферата:

Дата защиты реферата:

Студент гр. 8303

Почаев Н.А.

Студент гр. 8303

Перелыгин Д.С.

Студент гр. 8304

Алтухов А.Д.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В рамках указанной дисциплины предполагается создание веб-сайта и (или) приложения в команде с использованием технологии NoSQL. В виде предметной области данного проекта выбрано использование NASA API для предоставления информации об астероидах и метеоритах.

Найти исходный код и дополнительную информацию можно по ссылке:
<https://github.com/moevm/nosql2h21-nasa>

SUMMARY

In the framework of the mentioned discipline, it is supposed to create a website and (or) an application in a team using NoSQL technology. The subject area of this project is the use of NASA API to provide information about asteroids and meteorites.

Find the source code and more information at this link:
<https://github.com/moevm/nosql2h21-nasa>

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью данной работы является создание веб-сайта, предоставляющего функциональность по получению визуализированной информации (карты, графики, таблицы) по астероидам и метеоритам, известным NASA. Также данный сайт предоставляет функциональность по просмотру текущего положения МКС. Поддержано добавление новых данных, а также импорт и экспорт базы данных. Использованный стек: Node.js, Handlebars, MongoDB, Docker.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Требуется разработать user-friendly веб-сайт, реализующий и предоставляющий пользователю все заявленные функции. Представление данных должно быть наглядным и интерактивным. В качестве системы управления базами данных используется MongoDB.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. UI реализованного сайта

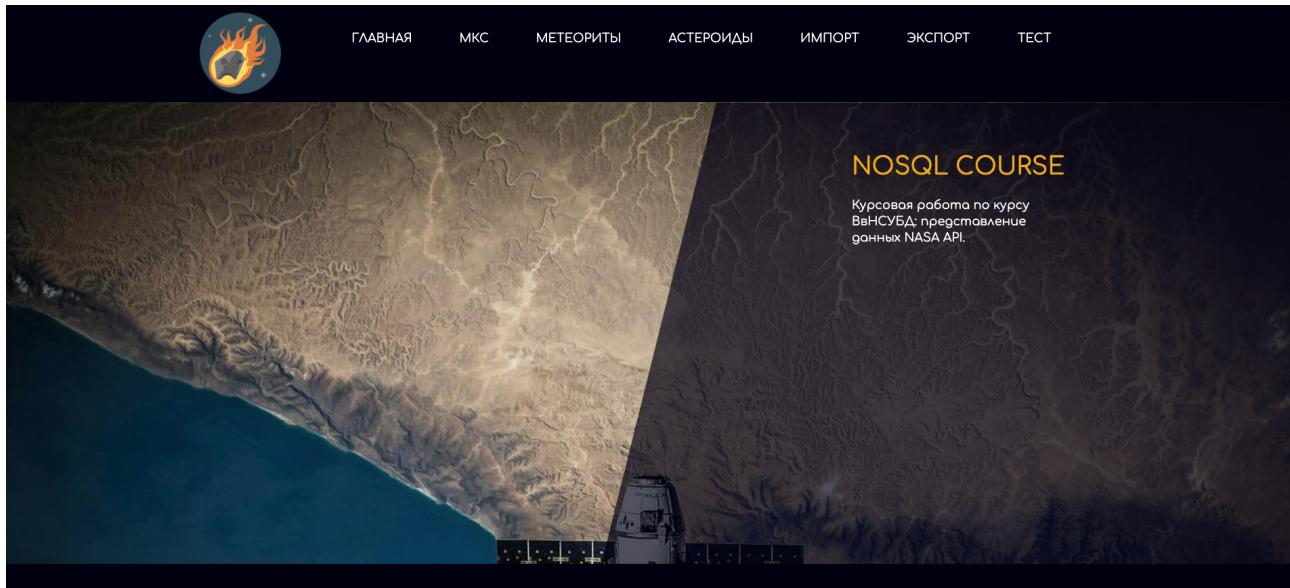


Рисунок 1 – Стартовая страница сайта

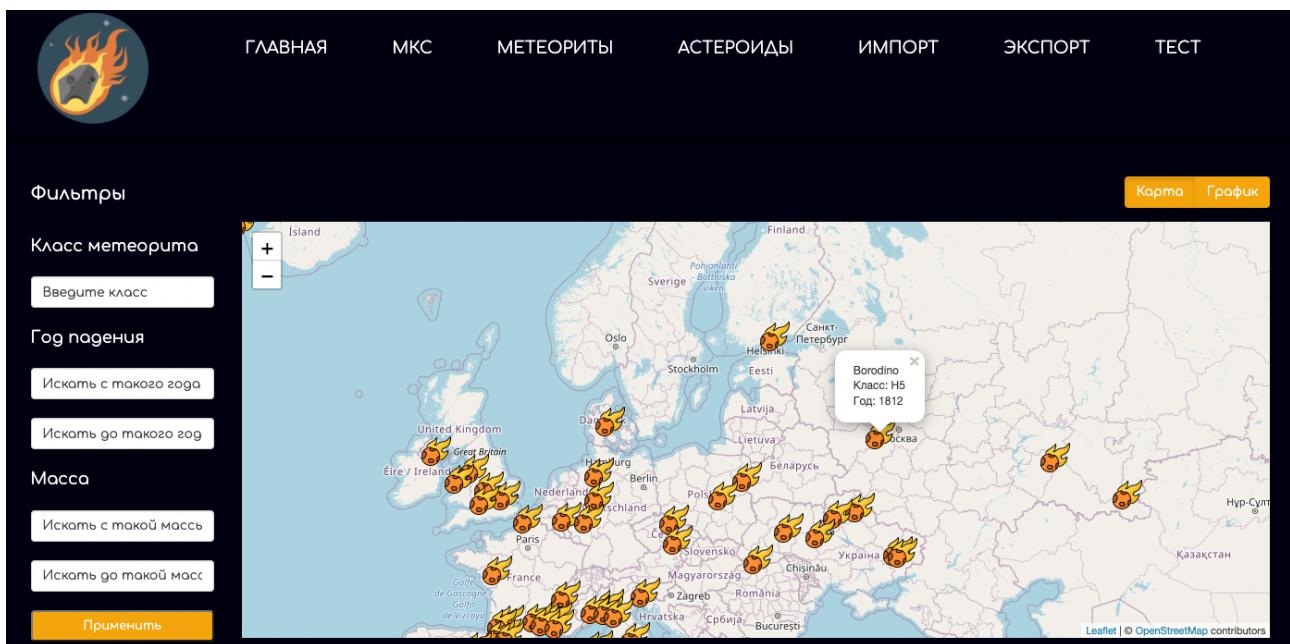


Рисунок 2 – Карта метеоритов с возможностью фильтрации

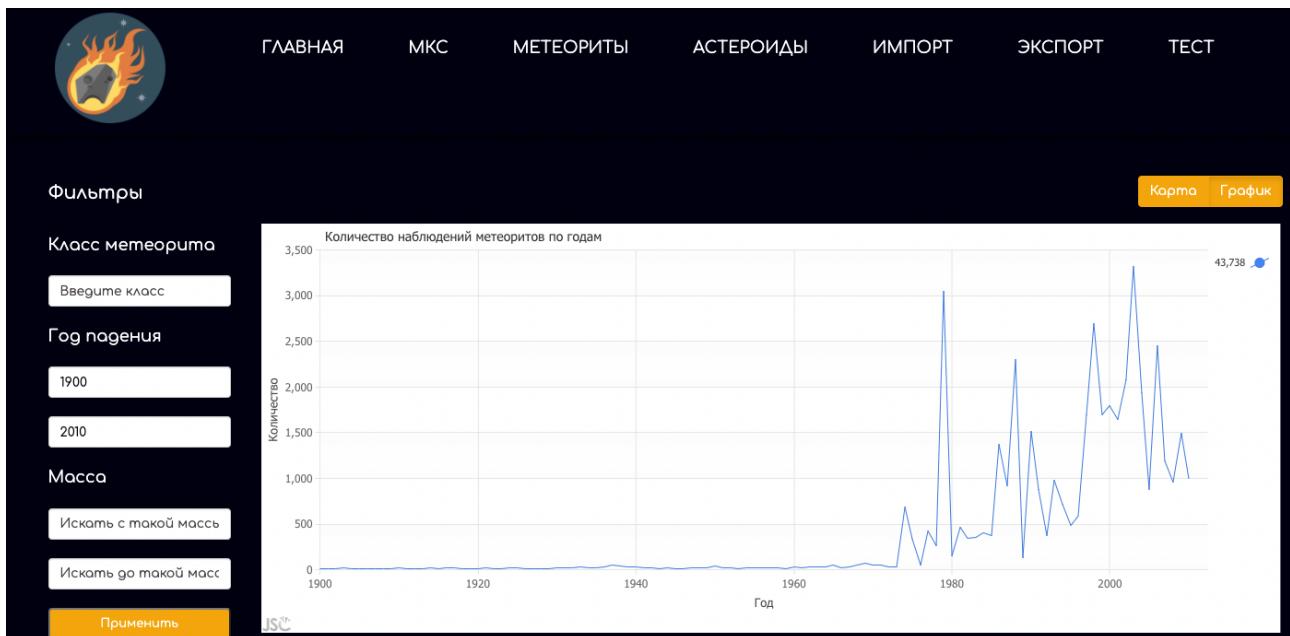


Рисунок 3 – График падения метеоритов по годам

ГЛАВНАЯ МКС МЕТЕОРИТЫ АСТЕРОИДЫ ИМПОРТ ЭКСПОРТ ТЕСТ

Добавить

Название
Введите название

Класс метеорита
Введите класс

Масса метеорита
Введите массу (г)

Год падения
Введите год падения

Широта
Введите широту

Долгота
Введите долготу

Применить

Название	Класс	Масса (г)	Год падения
Aachen	L5	21	1880
Aarhus	H6	720	1951
Abee	EH4	107000	1952
Acapulco	Acapulcoite	1914	1976
Achiras	L6	780	1902
Adhi Kot	EH4	4239	1919
Adzhi-Bogdo (stone)	LL3-6	910	1949
Agen	H5	30000	1814
Aguada	L6	1620	1930
Aquila Blanca	L	1440	1920
Aioun el Atrouss	Diogenite-pm	1000	1974

Рисунок 4 – Таблица всех метеоритов с возможностью добавления новых



ГЛАВНАЯ МКС МЕТЕОРИТЫ АСТЕРОИДЫ ИМПОРТ ЭКСПОРТ ТЕСТ

Фильтры	Название	Диаметр (км)	Риск столкновения	Перигейное расстояние (ае)	Апогейное расстояние (ае)	Дата первого наблюдения	Дата последнего наблюдения
Опасность столкновения	433 Eros (A898 PA)	16.84	Нем	1.1329725897	1.7831188684	1893-10-29	2020-09-03
	719 Albert (A911 TB)		Нем	1.1964517695	4.0807523593	1911-10-04	2020-02-27
	887 Alinda (A918 AA)	4.2	Нем	1.0628862956	3.8845875638	1918-02-03	2020-09-07
	1036 Ganymed (A924 UB)	37.675	Нем	1.2443035104	4.0851457382	1924-10-23	2020-09-16
Диаметр	1221 Amor (1932 EA1)	1	Нем	1.0839694978	2.7550268036	1932-03-12	2020-09-05
	1566 Icarus (1949 MA)	1	Да	0.1865002031	1.9698376456	1949-07-01	2020-09-08
	1580 Betulia (1950 KA)	5.8	Нем	1.126368444	3.2679316188	1950-05-22	2019-12-10
	1620 Geographos (1951 RA)	2.56	Да	0.8278019326	1.6635317949	1951-08-31	2020-03-05
	1627 Ivar (1929 SH)	9.12	Нем	1.1239578073	2.6025880828	1929-09-25	2020-06-22
	1685 Toro (1948 OA)	3.4	Нем	0.771514536	1.9636314073	1948-07-17	2020-09-05
	1862 Apollo (1932 HA)	1.5	Да	0.6470744134	2.2936164607	1930-12-13	2020-02-25
	1863 Antinous (1948 EA)	2.1	Нем	0.8886686397	3.628154422	1948-03-05	2020-08-09

Рисунок 5 – Таблица астероидов с полной информацией и возможностью фильтрации

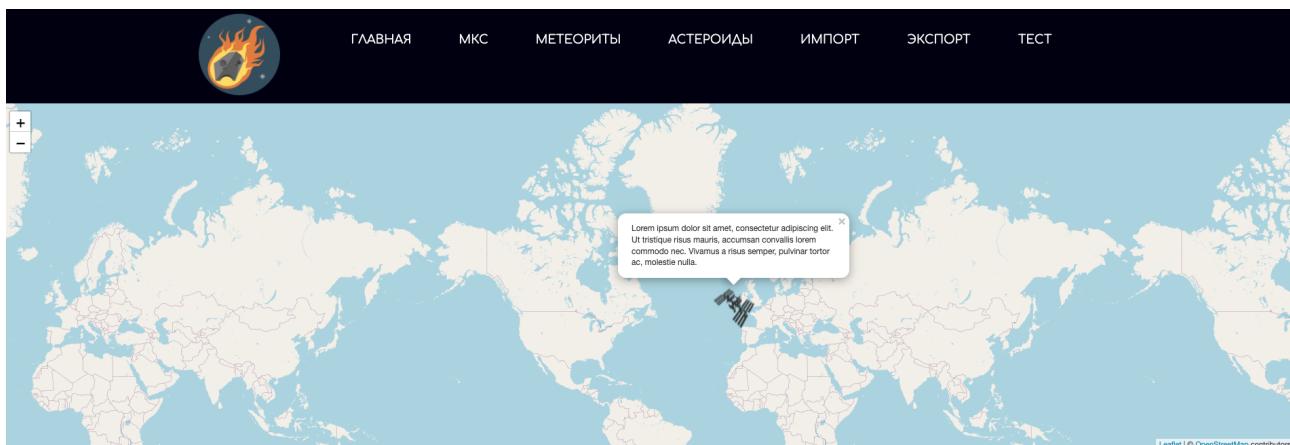


Рисунок 6 – Карта текущего местоположения МКС

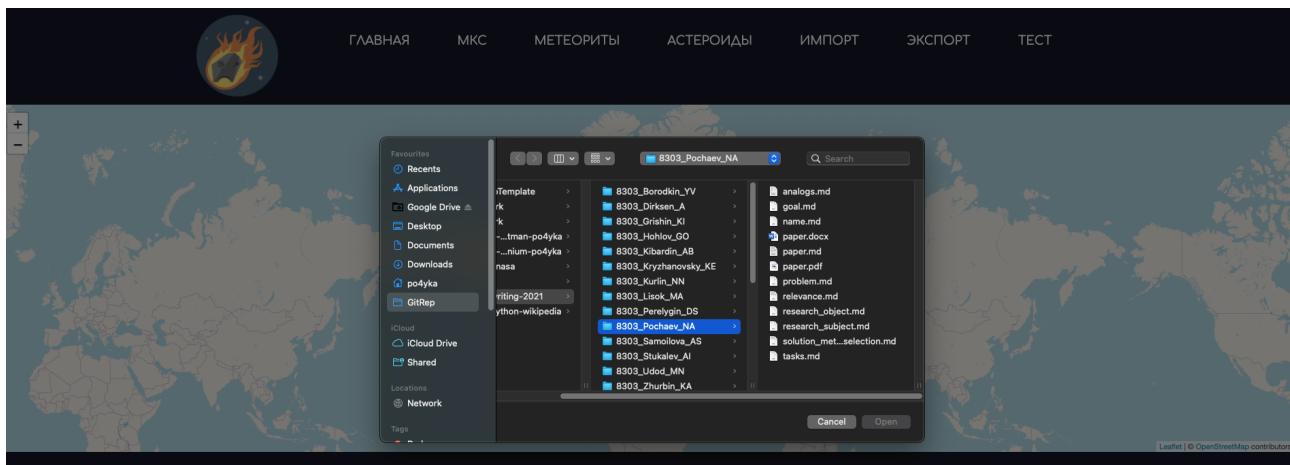


Рисунок 7 – Окно экспорта и импорта БД

2.2. Сценарии использования

Действующее лицо: Пользователь

1. Сценарий – «Просмотр основной предоставляемой информации»

Основной сценарий:

- a. Пользователь открывает главную страницу сайта
- b. Пользователь просматривает основную теоретическую информацию и предоставляемой информации
- c. Пользователь переходит в разделы, предоставляющие структурированную информацию об астероидах и кометах; о текущем местоположении МКС

Результат: пользователь узнает требуемую ему информацию

2. Сценарий – «Экспорт/Импорт»

- a. Пользователь нажимает на соответствующую функциональную кнопку импорта/экспорта в приложении и открывает системное диалоговое окно указания пути
- b. Пользователь инициирует выполнения импорта/экспорта БД на сервер

3. Сценарий – «Использование продвинутых методов фильтрации и получения статистики»

Предусловие: пользователь находится в одном из разделов – «Метеориты» или «Астероиды»

- a. В боковом меню пользователь выбирает параметр фильтрации и нажимает «Применить»
- b. График / таблица / карта претерпевают изменения и начинают отображать актуальную для выбранных параметров информацию

3. МОДЕЛЬ ДАННЫХ

3.1. Схема данных в SQL

Схема данных, как она бы выглядела при использовании стандартной реляционной БД приведена ниже на рис. 8.

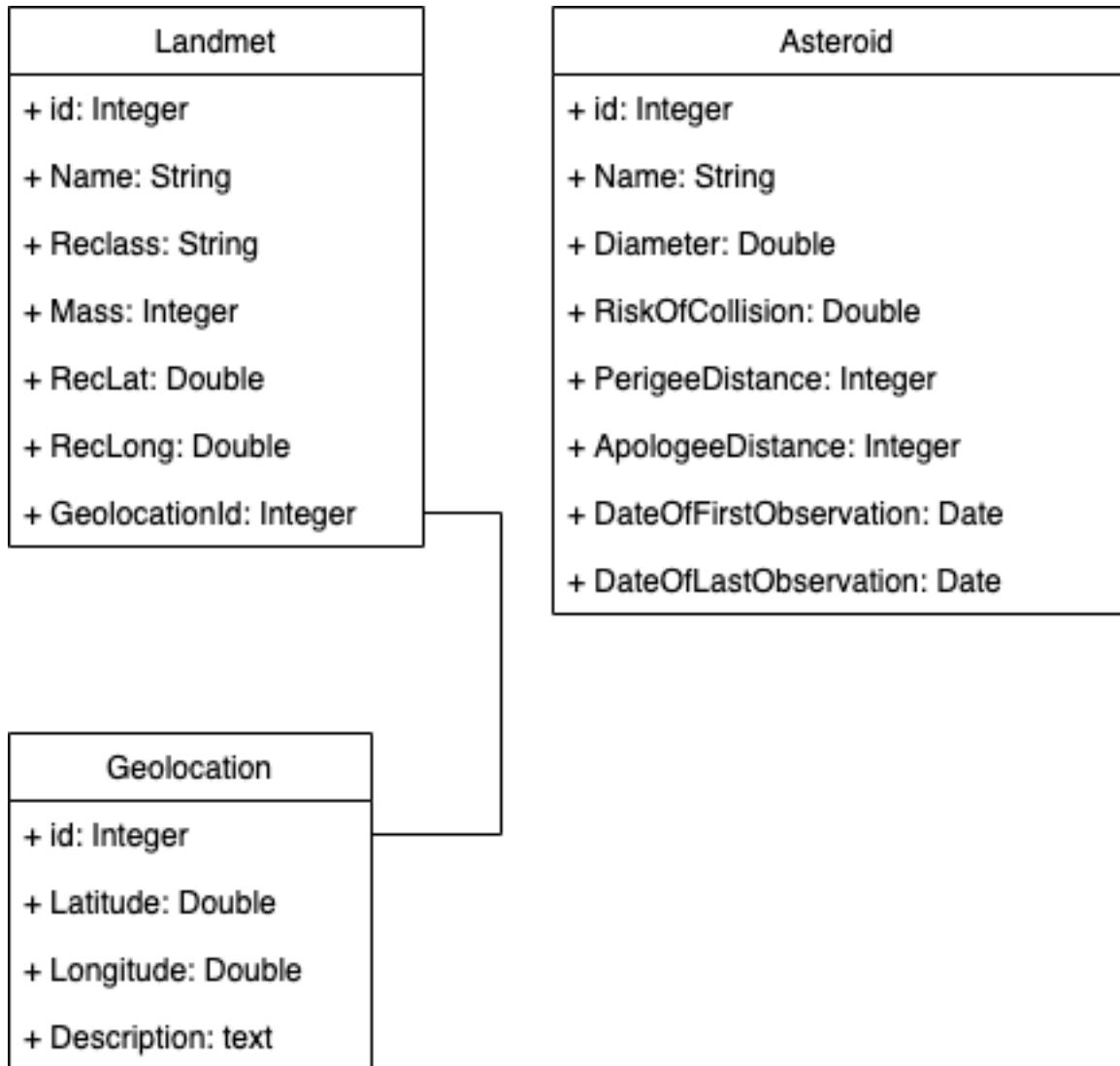


Рисунок 8 – Схема данных в SQL

3.2. Схема данных в NoSQL, сравнение с SQL

Схема данных в NoSQL БД (в контексте данной работы – MongoDB) представлена на рис. 9. Представление в формате JSON представлены на рис. 10-11.

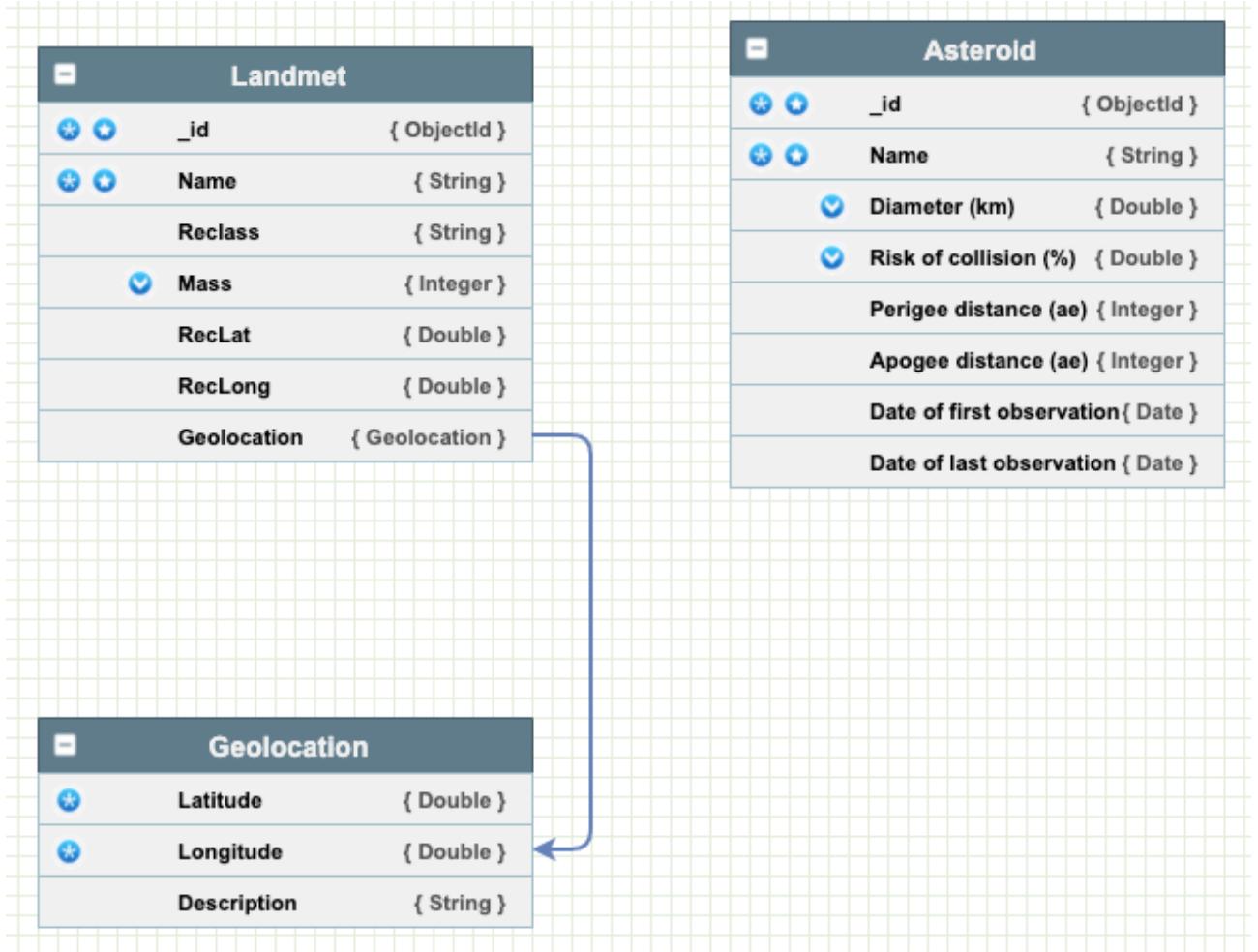
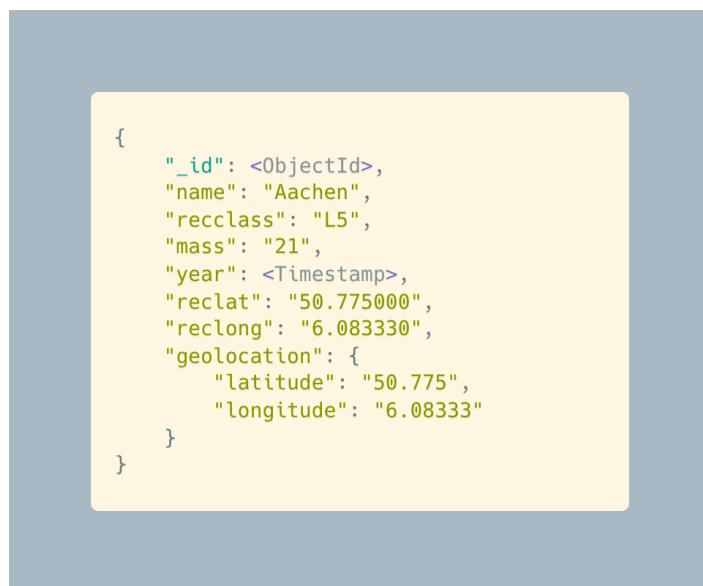


Рисунок 9 – Схема данных в NoSQL

- Landmet:
 - id - уникальный идентификатор;
 - Name - название / имя метеорита;
 - Reclass (class) - класс метеорита;
 - Mass - масса;
 - RecLat - широта нахождения;
 - RecLong - долгота нахождения;
 - GeolocationId - локация падения с широтой и долготой.
- Asteroid:
 - id - уникальный идентификатор;
 - Diameter - диаметр;
 - RiskOfCollision - риск столкновения;
 - PerigeeDistance - расстояние Перегиля;

- ApogeeDistance - апогейное расстояние;
 - DateOfFirstObservation - дата первого наблюдения;
 - DateOfLastObservation - дата последнего наблюдения.
- Geolocation:
 - id - уникальный идентификатор;
 - Latitude - широта падения;
 - Longitude - долгота падения;
 - Description - описание падения.



```
{
  "_id": <ObjectId>,
  "name": "Aachen",
  "recclass": "L5",
  "mass": "21",
  "year": <Timestamp>,
  "reclat": "50.775000",
  "reclong": "6.083330",
  "geolocation": {
    "latitude": "50.775",
    "longitude": "6.08333"
  }
}
```

Рисунок 10 – Схема данных метеорита в MongoDB



```
{
  "_id": <ObjectId>,
  "name": "2010 PK9",
  "recclass": "L5",
  "diameter": "21",
  "risk_of_collision": 35.5,
  "perigee_distance": 1000,
  "apogee_distance": 500,
  "date_of_first_observation": <Timestamp>,
  "date_of_last_observation": <Timestamp>,
  "year": <Timestamp>
}
```

Рисунок 11 – Схема данных астероида в MongoDB

Каждая сущность, как астероид, так и метеорит, представляют собой отдельные документы. Поле Geolocation у сущности Landmet является вложенным документом.

3.3. Список сущностей

В данном проекте осуществляется хранение и обработка данных для двух явлений: метеоритов и астероидов.

- Метеорит - крупные метеорные тела, которые падают на Землю.
- Астероид - малые планеты - твёрдые небесные тела, которые значительно меньше планет по размеру. Движутся по орбитам.

3.4. Оценка объёма информации

Рассмотрим сущность Landmet для наглядности (наличие вложенного документа). Пусть совокупность полей Name, Reclass, Mass, RecLat, RecLong занимает N памяти, M - количество упавших метеоритов, а вложенный документ Geolocation (поля Latitude, Longitude, Description) - K .

Тогда для хранения данных будет занято:

- MongoDB: $M * (N + K) = M * N + M * K$
- SQL: $M * N + * M * K$

Тогда, согласно [документации](#), $N = 32$ байт название (строка в UTF-8) + 64 байт масса + 32 байт класс + $32 * 2$ (координаты) = 160 байт. В реляционной БД результаты будут аналогичными. Таким образом, кол-во чистой памяти, занимаемой реляционной БД, не отличается от MongoDB.

Количество памяти, занимаемой MongoDB с учётом типа <ObjectId> (12 байт) и overhead (e_list + zero-termination) в 1 байт для каждого элемента = $12 + 5 + 160 = 177$ байт.

Количество памяти, занимаемой реляционной БД с учётом полей id (4 байта): $4 \text{ байта} * 4 \text{ id} * 2$ (с учётом ссылок) + 160 = 192 байт.

Таким образом, MongoDB выигрывает реляционной БД при учёте всей информации, хранимой в моделе (расчёты для одной записи / документа).

Формула роста памяти для MongoDB при увеличении кол-ва моделей будет иметь вид: $N * (12 + 5 + 160) = 177N$, а для SQL: $192N$.

Как следствие, использования MongoDB по памяти в данном случай будет эффективнее на $177/192$.

3.5. Запросы

- Запрос на поиск метеорита по id по всей коллекции (без учёта индексов) по местоположению:
 - MongoDB: $O(M)$
 - SQL DB: $O(2M^2)$, т.к. необходимо сделать 2 раза операцию JOIN на таблицу Geoposition.
- Запросы на текущий момент осуществляются стандартным путём. (WIP)

3.6. Исследование итоговой производительности

В ходе экспериментов с готовой системой измерены следующие характеристики работы приложения:

- время обработки запросов к БД;
- фактический объем хранимых данных.

Результаты измерений представлены в табл. 1-2 соответственно. И на графиках далее.

Таблица 1 – Зависимость кол-ва записей в БД и времени запроса

Кол-во записей	Время, с
2285	3,212599999
4570	5,19884
6855	6,370085
9170	8,119725
11455	9,66861
13740	10,097995

16025	12,52824
18310	13,633155
20595	13,9511
22880	13,999335
25165	15,42216
27450	17,568295
29735	18,37964
32020	21,001875
34405	22,514435
36590	23,777655
38875	25,0248
41160	28,460135
43445	30,5348
45730	31,76351

Таблица 2 – Зависимость кол-ва записей в БД и времени запроса

Кол-во записей	Объём памяти, Мб
2285	469377
4570	941020
6855	1410916
9170	1874556
11455	2377132
13740	2866904
16025	3371357
18310	3841239
20595	4291188
22880	4758899
25165	5229756
27450	5673780
29735	6103881
32020	6577215
34405	7065333
36590	7540538
38875	8011883
41160	8486672
43445	8961453

45730	9400795
-------	---------



Рисунок 12 – Зависимость времени выполнения запроса к БД от кол-ва записей



Рисунок 13 – Зависимость потребляемой памяти БД от кол-ва записей

Как из приведённых практических данных, спроектированная и реализованная БД, использованная в созданном веб-сайте, полностью соответствует заявленным положениям о теоретической работе по потребляемой памяти и скорости выполнения запросов – сложность и рост являются линейными.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы была спроектирована и разработана БД NoSQL с использованием MongoDB, полностью соответствующая своей области применения и производительная, согласно теоретическим подсчётом. Также был разработан user-friends сайт, позволяющий пользователю видеть подробную структурированную информацию о метеоритах и астероидах, предоставляемую NASA.